

SYSTEM AND METHOD FOR MAPPING INTERFACE FUNCTIONALITY TO CODEC FUNCTIONALITY IN A PORTABLE AUDIO DEVICE

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Patent
5 Application No. 60/241,218 filed October 13, 2000, where this provisional
application is incorporated herein by reference in its entirety.

TECHNICAL FIELD

10 The present invention is related generally to portable audio devices
and, more particularly, to a system and method for mapping interface functionality
to CODEC functionality in a portable audio device.

BACKGROUND OF THE INVENTION

15 Portable audio devices have evolved from large cumbersome analog
tape players to highly miniaturized digital storage devices. Early portable audio
devices were typically in the form of analog tape players that sequentially played
musical selections (or other audio presentations). For example, a prerecorded audio
tape could be purchased by the user and sequentially played in a portable tape
player. However, the user had no control over the sequence of play other than to
20 stop the playing and manually fast forward or rewind to skip over one or more
selections.

With the advent of portable digital devices in the form of compact
disk (CD) players, the user has additional flexibility in the selections of songs from
a CD. For example, some CD players permit the user to manually enter the
sequence of musical tracks that will be played rather than play the musical tracks in
25 a predetermined sequence from start to finish. Alternatively, some CD players also
include a "random" mode in which musical tracks are randomly selected. However,

the CD players described above are still limited to the selection of musical tracks on a single CD. Digital musical devices have been designed to eliminate all moving parts. These devices incorporate solid state memory storage technology and utilize digital processing capabilities, such as data compression, to minimize data storage requirements. A popular musical format, known as Motion Pictures Expert Group layer 3 (MPEG-2 layer 3) defines a digital musical format that plays "near-CD quality" music from a relatively small digital file as compared with the original digital file stored on a CD. Using known data compression techniques, the data structure defined by MPEG-2 layer 3, sometimes abbreviated as MP3, is approximately one tenth the size of a comparable data file on a CD.

A further advantage of a portable digital audio device is that it is capable of playing virtually any audio data file. In the example presented above, music data files that contain multiple musical tracks may be stored in the portable digital device and played back. However, other data files, such as audio books, may also be played back by the portable digital audio device. The type of signal processing for different audio data files varies greatly. For example, music data files require playback at significant data rates to achieve high quality stereo audio output signals while audio books require significantly lower data rates and need not be in stereo. The different data processing requirements for different data file types requires different user interaction with the portable digital device. This requires a large number of buttons for operation by the user, or significant learning by the user as to the functionality of buttons in different operational modes. Accordingly, it can be appreciated that there is a significant need for a system and method that automatically alters functionality of a user interface based on a type of data being processed. The present invention provides this and other advantages, as will be apparent from the following detailed description and accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a functional block diagram of an exemplary embodiment of the present invention.

5 Figure 2 is a top plan view of one embodiment of the present invention.

Figures 3-8 are various screen displays illustrating the operation of the present invention in various data entry and editing modes.

Figures 9-11 together form a flow chart illustrating the operation of the system of the present invention.

10 Figure 12 is a functional block diagram illustrating the flow of data between elements of a portable digital audio device illustrating another inventive aspect thereof.

Figures 13 and 14 together form a flowchart illustrating the operation of the system of Figure 12.

15 DETAILED DESCRIPTION OF THE INVENTION

The present invention automatically selects a proper data processing mode and display based on the type of data file selected for processing by a user. When a user selects a data file to play, the system automatically determines the type of data file, selects the proper data processing component, and configures the user
20 interface and display for proper operation with the selected data file.

The present invention is embodied in a system 100, illustrated in the functional block diagram of Figure 1. The system 100 includes a central processing unit (CPU) 102 and a memory 104. The CPU 102 may be implemented using a device, such as the ARM 7209 from Cirrus Logic or other processor designed for
25 operation as an MP3 player. However, those skilled in the art will appreciate that the CPU 102 may be implemented using any convenient processor, such as a microprocessor, embedded controller, digital signal processor (DSP) or the like. The present invention is not limited by the specific form of the CPU 102. The

memory 104 may typically include both random access memory (RAM) and read-only memory (ROM). In one embodiment, the ROM portion of the memory 104 may be implemented using a flash program memory or a NAND flash memory. In addition, the memory 104 includes a basic input output system (BIOS), which
5 contains instructions that allow the CPU 102 to communicate with various peripheral devices.

In addition, the system 100 includes a display 108. In an exemplary embodiment, the display 108 is implemented as a liquid crystal display (LCD) to reduce overall power consumption. In one example, the display 108 may be a 240
10 by 160 pixel LCD subsystem, such as may be commercially purchased from a number of vendors. The display 108 may conveniently provide instructions to the user as well as programmable functions that may be context-sensitive. For example, when playing a music signal, the display 108 may provide commands associated with music playing, song information, and the like. For example, the display 108
15 may show the data sampling rate and number of kilobytes (Kb) in a particular data file. The display 108 may also include other information, such as power status, startup information, and the like.

The system 100 also includes an input device 110. The input device 110 may be implemented as a series of electromechanical switches using
20 conventional techniques. Alternatively, the input device 110 may be implemented in conjunction with the display 108 to provide a touch-sensitive display. A touch-sensitive display advantageously minimizes the need for electromechanical switches and further provides labels on the display that may be readily altered to accommodate variations in the implementation of the system 100. Alternatively, the
25 input device 110 may comprise both electromechanical switches and a touch-sensitive display. Electromechanical switches and touch-sensitive displays are known in the art and need not be described in further detail herein. However, the present invention is not limited by the specific form of the input device 110.

As those skilled in the art can appreciate, the data representing the audio signal is in the form of digital samples. The digital data must be converted to analog form to produce a useful signal for the user. The system 100 includes a coder/decoder (CODEC) 114. The CODEC 114 is also sometimes referred to as a
5 “compressor/decompressor” because the digital data samples are usually stored in a compressed form and are decompressed for playback. The CODEC 114 accepts a digital data stream and converts it to a representative analog signal. Different commercial CODECs are available for audio applications. Some CODECs, such as a code excited linear prediction (CELP) CODEC, developed in 1985 by Schroeder
10 and Atal, is designed for operations at relatively low frequencies and thus is particularly useful as a speech CODEC. Other forms of speech CODECs include adaptive delta modulation (ADM), pulse code modulation (PCM) and adaptive differential pulse code modulation (ADPCM).

Other forms of CODECs are designed for operation at higher data
15 sampling rates and are thus useful for music applications. These music CODECs include MPEG or MP3 CODECs, G2 format, developed by Real Networks, Enhanced Perception Audio Decoder (ePAC), developed by Lucent, AC3 algorithm, which is a modified version of PCM, and Windows Media Audio (WMA), developed by the Microsoft Corporation. Some formats, such as the G2 format,
20 may be used for both music and voice. Although the examples illustrated herein are directed to MP3 music format, those skilled in the art will recognize that the CODEC 114 illustrated in Figure 1 may be satisfactorily implemented using any of the known CODEC technologies for either speech applications, music applications, or both. Thus, the present invention is not limited by the specific implementation of
25 the CODEC 114.

In an MP3 environment, the digital data is provided to the CODEC 114 using an I²S bus. The I²S bus is a high speed serial bus that is well known to those of ordinary skill in the art. As such, implementation details of the I²S bus need not be provided herein. The CODEC 114 receives the data on the I²S bus and

converts it from digital data form to analog data. An analog amplifier 116 has an input terminal coupled to the output of the CODEC and receives the analog signal thereon. The amplifier 116 provides the necessary amplification and drive capability to power an audio output device 118, such as a pair of headphones. It should be noted that in a typical implementation, the output of the amplifier 116 is coupled to a standard 1/8 inch phone jack (not shown). The headphones 118 plug into the phone jack.

The system 100 also includes a buffer 124 that receives and temporarily stores digital data and provides the digital data to the CODEC 114. As will be discussed below, the buffer 124 receives data from a storage device 126. The buffer 124 may be a stand-alone device, or may be a portion of the memory 104. The use of the buffer 124 in optimizing the response of the storage device 126 will be discussed below.

The storage device 126 is typically implemented as a spinning media device, such as a micro-drive, click drive, or the like. The storage device 126 has a controllable motor (not shown) that is only enabled when the system 100 requires a data transfer to or from the storage media. The optimization of the storage device 126 includes a determination of when to start the motor on the storage device to allow it to come up to full speed, and how long to maintain power to the motor so as to transfer the desired amount of data from the storage media to the buffer 124.

Those skilled in the art will recognize that the storage device 126 is an optional component and may be eliminated without adversely affecting the operation of the present invention. A number of portable audio devices contain no storage device 126, but rely solely on the memory 104 to store the musical tracks. For the sake of completeness, the buffer 124 and storage device 126 are described herein. The buffer 124 is implemented in the system to optimize data transfer from the storage device 126. Although it is beyond the scope of the present invention, the buffer 124 may be allocated into a large number of buffer portions with one of the buffer portions being actively used to transfer data to the CODEC 114 while the

remaining buffer portions are available for data transfer from the storage device 126. If the system 100 is implemented without the storage device 126, the buffer 124 may also be eliminated without adversely affecting the operation of the system. In this implementation, the musical track data is transferred directly from the memory 104 to the CODEC 114. Because the memory 114 is a solid state memory, data transfer rates are sufficiently high to accommodate satisfactory data transfer to the CODEC so as not to cause interruptions in the generation of output data.

The system 100 also may include an optional input/output (I/O) interface 130. The system 100 may include any conventional form of I/O interface and may typically include a serial interface and/or a universal serial bus (USB) interface. The operation of a serial interface and USB interface are well-known in the art and need not be described in greater detail herein. Although illustrated as a single I/O interface 130, those skilled in the art will recognize that the I/O interface 130 is intended to illustrate the function of one or more conventional interfaces.

A power supply 132 provides power to all of the components of the system 100. In an exemplary embodiment, the power supply 132 comprises two or more AAA batteries. A voltage regulator (not shown) in the power supply 132 provides a regulated voltage of approximately 3.1 VDC. The power supply 132 may also include provisions, such as an external power supply jack 170 (see Figure 2), to permit the introduction of power from an external source, such as a cigarette lighter in an automobile, or the like.

The system also includes a data structure 134 to store data related to user-generated playlists and associated data. In one embodiment, the data structure 134 may be implemented as a database. However, those skilled in the art will recognize that any convenient form of known data structure will operate satisfactorily with system 100. Furthermore, the data structure 134 may be a portion of the memory 104 or a stand-alone data storage element. The present invention is not limited by the specific form in which the data structure 134 is implemented.

The various components of the system 100 are coupled together by a bus system 138. The bus system 138 may include a data bus, control bus, the I²S bus, a memory bus, and the like. However, for the sake of simplicity, these various buses are illustrated in Figure 1 as the bus system 138.

5 Some of the elements illustrated in the functional block diagram of Figure 1, such as the CODEC 114, may be elements that are actually implemented by the CPU 102 using computer instructions stored in the memory 104. As is known to those of skill in the art, the CODEC 114 is typically a software data processing element. However, it is illustrated in the functional block diagram of
10 Figure 1 because it performs a separate data processing function. As will be discussed in greater detail below, the system 100 may implement a number of different CODECs, one of which is selected as the CODEC 114, based on the type of data to be processed. For example, the CODEC 114 may be an MP3 CODEC if the data file to be processed is a music data file. In contrast, the CODEC 114 may
15 be a different implementation (*e.g.*, the CELP CODEC) if the data file is a speech data file, such as an audio book.

 The system 100 is intended for portable operation. The various components described above are typically implemented as one or more integrated circuits on a printed circuit (PC) board (not shown). The PC board power supply
20 132, display 108, input device 110, and other components of the system 100 are enclosed in a case or housing 150, as illustrated in Figure 2. As further illustrated in Figure 2, the input device 110 comprises a four-button key pad assembly 152, a two-button key pad assembly 154, and an optional joystick 156. The four-button key pad 152 may be conveniently configured to function in a manner similar to
25 well-known hand-held electronic games. Alternatively, the four-button key pad 152 can be replaced with a membrane (not shown) to permit the operation of four hardware buttons in a manner similar to a top hat switch on a joystick wherein one or two of the buttons may be activated to provide eight unique switch settings. In yet another alternative, the four-button key pad 152 or the two-button key pad 154

could be replaced with a position-sensing membrane, such as a touch pad commonly used in laptop computers. Those skilled in the art will recognize that other configurations may also be used for the input device 110. As will be described in greater detail below, the display 108 may conveniently comprise touch-sensitive display technology that will allow readily alterable configurations for control buttons that will correspond with the particular data shown on the display 108. A power switch 158 may be conveniently installed in the side of the housing 150 to allow the user to turn the system on and off.

When power is first applied to the system 100, the display 108 may be configured to illustrate a main menu, such as illustrated in the screen display 160 of Figure 3. The screen display 160 may include a series of icons 164, such as a jukebox icon 166, a player icon 168, and the like. In addition to icons 164, the screen display 160 may include touch-sensitive programmable controls, such as a "Scroll Up" control button 172, a "Selection" control button 174, a "Scroll Down" control button 176 and an "Exit" control button 178. The operation of a touch-sensitive screen to implement these buttons are well known and need not to be described in any greater detail herein. Furthermore, the operation of the buttons, such as the Scroll Up button 172 and the Scroll Down button 176 are well known in the art and need not be described in detail. Activating the Scroll Up button 172 or the Scroll Down button 176 will cause the display to highlight a different one of the icons 164. When the desired icon is highlighted, such as by reverse video or other conventional technique, the user may activate the selection button 174 to activate the selected function.

Figure 4 illustrates a sample screen display 182 shown by the system in response to the activation of the jukebox icon 166 and the selection of one playlist. As previously noted, the system 100 supports a plurality of different playlists. The screen display 182 comprises a playlist title portion for a playlist title display 184 to permit the user to readily identify the selected playlist. The user may simply activate the playlist to play musical tracks in the predetermined sequence

shown in the playlist by pressing the Selection control button 174. When a display list is first shown on the display 108, the first entry in the playlist may be automatically selected and indicated using, by way of example, reverse video. The user may also scroll through the selected playlist using a scroll bar 190 in a well-known fashion or, alternatively, simply by touching the touch-sensitive display 108 at a point corresponding to the desired musical track. The system 100 may also be configured to allow the user to scroll through the selected playlist using the Scroll Up button 172, a Scroll Down button 176, and the Selection control button 174 in the manner described above to select a musical track out of the sequence illustrated in the playlist.

The user may also control the operation of the system 100 to open or edit playlists, or create new playlists using additional programmable control buttons 192 on a predetermined portion of the touch-sensitive display 108. The Programmable control buttons 192 may comprise buttons such as a "Open" control button 194, an "Edit" control button 196 and a "New" control button 198. The Open control button 194 may be used to display a number of different playlists and permit the user to select from one of the displayed playlists in the manner described above. That is, the user may activate the scroll bar 190 or the Scroll Up button 172, the Scroll Down button 174, and the like, to navigate through the displayed playlists. As the displayed playlists scroll up or down the display 108, a selected display list is shown in a highlighted fashion, such as reverse video. The user opens the selected playlist using the Selection control button 174 or another one of the convenient Programmable control buttons 192. The user may edit a selected playlist by selecting the Edit control button 196.

The user may edit an existing playlist by activating the Edit control button 196. Activation of the Edit control button 196 will cause the system 100 to display the names of already established playlists. The user may manipulate through the lists of playlists using, by way of example, the scroll bar 190 to select the desired playlist. When the desired playlist has been selected, the display 108

will indicate the musical tracks already selected in the playlist, as illustrated in Figure 4. In an exemplary embodiment, the first musical track in the playlist is highlighted using, by way of example, reverse video. The user selects a particular musical track in the manner described above. The user can edit a selected musical track, to correct misspellings or other information, delete an existing musical track from the current playlist, or add additional musical tracks to the selected playlist using conventional editing techniques. The user exits the edit mode by activating the Exit control button 178.

In addition to editing an existing playlist, the user may elect to create a new playlist by activating the New control button 198. When the user activates the New control button 198, the display 108 may be configured to show all musical tracks currently stored in the memory 104. The user may scroll through the list of musical tracks using conventional controls, such as the scroll bar 190. As the user scrolls through the list of musical tracks, a selected musical track may be highlighted using, by way of example, reverse video. Other conventional techniques, such as bold video, underlined text, an asterisk or other indicator, may also be used to indicate the selected musical track. To enter a selected musical track into the new playlist, the user may activate the Selection control button 174. The user may scroll through the displayed list of stored musical tracks and select other musical tracks in the manner described above to thereby enter them into the playlist. When the playlist is completed, the user may exit the data entry mode by selecting the Exit control button 178. Thus, the system 100 has provided the user with a simple technique for creating music playlists.

When a playlist or individual musical track has been selected, that selection may be played by activating the Selection control button 174 or a special control button, such as a "Play/Pause" button 200. When a selected musical track begins to play, the touch-sensitive display 108 may be reprogrammed to show a screen display 202, illustrated in Figure 5. The touch-sensitive display 108 has also been changed such that the control buttons perform different functions relevant to a

media player. For example, the Scroll Up control button 172 and Scroll Down control button 174 may now be used to control the volume. A graphical representation 204 may provide visual cues to the user as to the volume level. The programmable control buttons 192 may now comprise a Fast Forward button 206 and Rewind button 208 to advance or rewind within the selected musical track. A Skip Forward button 210 may be used to automatically advance to the next musical track in the playlist while a Skip Rewind button 212 may be activated to rewind to the beginning of the current musical track if activated once and rewound to the beginning of the previous musical track in the playlist if activated twice within a short period of time. In addition, the Play/Pause control button 200 may be used in the manner previously described.

In addition to control buttons, the display screen 202 can provide user information, such as the currently selected function 220, a title 222, an artist name 224, and a track selection 226. Other information, such as an elapsed time 230, stereo indicator 232, sample rate indicator 234, and bandwidth indicator 236 may also be provided on the display screen 202. In addition, an exemplary embodiment of the system 100 may include a graphical equalization display 238 to indicate the relative power of signals at different frequency bands. Those skilled in the art will recognize that numerous variations are possible with the present invention. For example, the graphical equalization display 238 can be eliminated and replaced with other information, such as metatags indicating categories or other identifier tags that correspond to the selected musical track.

One convenient aspect of on-screen programming using the display 108 is that many configurations are possible. An alternative configuration of the media player is illustrated in Figure 6 where the programmable controls 192 have a different appearance, but perform the same functions as previously described with respect to Figure 5. In addition, the Scroll Up control button 172, Scroll Down control button 176 and Exit button 178 have a different appearance in the display screen 202 of Figure 6, but perform identical functions to those described above

with respect to the corresponding buttons in Figure 5. In Figure 6, the selection control button 174 has been replaced with a Repeat control button 240 to permit the user to repeat a selected musical track or selected musical playlist. Other programmable features, such as random selection of musical tracks within a playlist, and the like may also be readily provided using the touch-sensitive display 108.

Although the operation of the system 100 has been described with respect to buttons on the touch-sensitive display 108, similar control of the system may be accomplished using, by way of example, the four-button key pad 152 (see Figure 2) and the two-button key pad 154. Essentially, the buttons of the four-button key pad 152 and two-button key pad 154 are mapped into the functions described above with respect to the Programmable control buttons 192 and the control buttons 172-178. The operation of the four-button key pad 152 and two-button key pad 154 is within the scope of knowledge of one of ordinary skill in the art and thus, need not be described in greater detail herein.

The operation of the system 100 to open, edit, or create playlists has been previously described. In addition to selection of musical tracks by title, the system 100 advantageously allows the selection of musical tracks using metatags. In an exemplary embodiment, the system 100 creates the data structure 134 (see Figure 1) to store metatags corresponding to musical tracks stored in the memory 104 (see Figure 1). The data structure or database 134 may be part of the memory 104 (see Figure 1) or a separate data storage element. Those skilled in the art will recognize that any one of a number of well-known data structures may be satisfactorily used to implement the data structure described herein. For the sake of convenience, the data structure 134 will be subsequently described as a database. However, the present invention is not limited by the specific implementation of a data structure to store metatags.

A number of different data elements may be used as metatags. For example, the artist's name, song title, album title, date, copyright, or any other information associated with a musical track can be potentially used as a metatag. In

an exemplary implementation, the user may elect to create a new playlist by activating the New control button 198 (see Figure 4) using metatags to describe the desired musical tracks. In this example, illustrated in Figure 7, the display 108 shows a screen display 250 that lists a series of possible metatags for selection by the user. In an exemplary embodiment, the first metatag in the list of metatags is automatically selected. The user may scroll through the list using, by way of example, the scroll bar 190 to select a desired metatag, as illustrated in Figure 7. As noted above, the system 100 can automatically generate a playlist based on the user-selected metatag or provide a list of musical tracks that match the selected metatag for display and subsequent manual selection by the user. For example, if the user selected the metatag "Artist," the system 100 would permit the user to enter the name of a desired artist or, alternatively, will display the artist name for all musical tracks stored in the memory 104 (see Figure 1). When the user selects a desired artist, the system may automatically generate the playlist and include all songs stored in the memory 104 that have a metatag corresponding to the user-selected artist name. Alternatively, the system 100 can display all musical tracks whose metatag corresponds to the user-selected artist name and thereby permit the user to manually select which musical tracks will be added to the playlist.

In addition to the metatags discussed above, other metatags, such as musical genre may be used as a metatag. For example, songs may be classified as "Rock," "Blues," "Rap," and the like. If the user selects a particular metatag, the system 100 accesses the database to determine which musical tracks stored in the memory 104 (see Figure 1) correspond to the selected metatag. If the user selects genre as the desired metatag, the system 100 may generate a screen display 252 on the display 108, as illustrated in Figure 8, to list the various musical genre for musical tracks stored in the memory 104. As noted above, the first item in the list may be automatically selected and the user may alter the selection using, by way of example, the scroll bar 190. In the example illustrated in Figure 8, the user-selected musical genre is "Blues." The user may activate the selection using the Selection

control button 174. Once a particular genre, such as Blues, has been selected, the system 100 may search the data structure 134 (see Figure 1) and automatically generate a playlist containing the musical tracks stored in the memory 104 whose metatags match the selected musical genre (*i.e.*, Blues). Alternatively, the system
5 100 may search the data structure 134 and create a list of all musical titles stored in the memory 104 whose metatag matches the selected musical genre. The list may be shown on the display 108 to permit subsequent manual selection by the user.

It should be noted that each musical track may have a number of different metatags to easily enable the user to search the data structure and automatically generate playlists. The association of musical tracks with multiple
10 metatags makes it easier for the user to search for desired musical tracks. In certain cases, a musical track may appear in more than one category. For example, certain musical tracks may be considered to belong to multiple genre, such as “Rock” and “Popular.”

15 In an alternative embodiment, the system 100 permits searching by multiple metatags. For example, the user may wish to search the data structure 134 for musical tracks that match metatags for both artist name and a particular date. In another example, the user may wish to select a particular musical genre, such as “Rock” and date to automatically generate a musical playlist of rock songs prior to a
20 user-selected date.

The operation of the invention is illustrated in the flowchart of Figures 9-11. At a start 300, illustrated in Figure 9, it is assumed that the system is under power or has just been turned on by the user. In step 302, the system 100 shows the main display, such as illustrated in Figure 3. In decision 304, the system
25 determined whether the user has selected the jukebox function. If the user has not selected the jukebox function, the result of decision 304 is NO. In that event, the system moves to step 306 and executes the selected function, such as displaying a contact list of user-entered names, addresses and telephone numbers. These

additional functions are beyond the scope of the present invention and will not be discussed in greater detail herein.

If the user has selected the jukebox function, the result of decision 304 is YES. In that event, the system 100 queries the data structure 134 and extracts the titles of all existing playlists and, in step 308, the existing playlists are shown on the display 108 (see Figure 1). In decision 310, the system 100 determines whether the user has activated one or more buttons to select a playlist. If the user has selected a playlist for play, the result of decision 310 is YES and, in step 312, the system plays the selected playlist by transferring data from the buffer 124 (or the memory 104) to the CODEC 114 in a conventional fashion. As previously noted, the musical tracks of the selected playlist may be played sequentially in the sequence originally specified by the user when creating the playlist, in a new sequence specified by the user at the present time, or in some other fashion, such as random selection.

If the user has not selected a playlist to play, the result of decision 310 is NO. In that event, in decision 314, the system 100 determines whether the user has selected a playlist for editing. If the user has selected a playlist for editing, the result of decision 314 is YES and the system enters an edit mode, described in the flowchart of Figure 10. If the user has not selected a playlist for editing, the result of decision 314 is NO. In that event, the system determines, in decision 316, whether the user has activated one or more buttons to create a new playlist. If the user has activated one or more buttons on the system 100 to create a new playlist, the result of decision 316 is YES and, the system enters a data entry mode illustrated in Figure 11. If the user has not elected to create a new playlist, the result of decision 316 is NO and, in step 320, the system ends the control function operation and, in one example, may return to display the main menu in step 302. Those skilled in the art will recognize that a number of different possible flowcharts may be implemented by the present system. For example, the system 100 may return to decision 310 until the user selects an operation. In addition, the activation of other buttons, such as a main menu button (not shown) may be used to exit the control

function process and return to the main display in step 302. The flowchart of Figures 9-11 are intended simply as an illustration of possible control flow to create, edit, and play selected playlists. The present invention is not limited to the specific processing sequence illustrated in the flowcharts of Figures 9-11.

5 As previously stated, the user may activate one or more of the buttons on the system 100 to edit a selected playlist. If the user has elected to edit a selected playlist, the result of decision 314 in Figure 9 is YES. In that event, the system 100 moves to decision 330, illustrated in Figure 10, to determine whether the user has elected to alter a selected track. If the user has elected to alter a selected track, the
10 result of decision 330 is YES. In step 332, the system displays stored data about the selected track and may further display a keypad (not shown) for user operation to change selected data. For example, the user may wish to edit the title of a musical track to correct a typographical error from a previous entry. The user can highlight the selected data element (*e.g.*, the title) and activate the edit control button 196 (see
15 Figure 4). The user can operate the touch-sensitive display 108 to enter a new title. The altered data will be displayed and stored in subsequent steps described below.

If the user has not elected to alter a selected track, the result of decision 330 is NO. In that event, the system 100 moves to decision 336 to determine whether the user has activated one or more keys to delete a selected track
20 from the playlist. If the user has elected to delete a track from the playlist, the result of decision 336 is YES. In that event, in step 338, the system 100 deletes the selected track and the newly edited playlist is updated and stored in steps described below. The system 100 also checks to see if the user wishes to perform more edits, as will be described in greater detail below. If the user has not activated one or
25 more buttons on the system 100 to delete a musical track from the playlist, the result of decision 336 is NO.

In decision 340, the system 100 determines whether the user has activated one or more buttons on the system 100 to add a new musical track to an existing playlist. If the user has elected to add a new musical track to the playlist,

the result of decision 340 is YES. In that event, in step 342, the system 100 displays a list of all musical tracks that may be stored in the memory 104 (or the optional storage device 126). In step 344, the user selects the desired musical track to the selected playlist in the manner described above. In an exemplary embodiment, a musical track that may be stored on the optional storage device 126 may be relocated to the memory 104. Following the selection of the stored musical track in step 344, the system 100 returns to decision 340 to determine whether additional new tracks will be added to the selected playlist.

If no additional musical tracks are to be added to the existing playlist, the result of decision 340 is NO and the edit operation. Following the completion of the selected edit operation, such as altering the selected track in step 332, deleting a selected track in step 338, or adding selected tracks in steps 342-344, the system 100 moves to decision 350 to determine if the user wishes to perform additional edit operations on the selected existing playlist. If the user does not wish to end the current editing session, the result of decision 350 is NO and the system may return to decision 330 to permit additional editing of one or more tracks in the existing playlist.

If the user wishes to end the editing session by activating, by way of example, the Exit control button 178 (see Figure 4), the result of decision 350 is YES. In that event, in step 352, the system 100 updates the existing playlist to include all edits performed by the user and, in step 354, the system stores the newly edited playlist. As previously discussed, the edited playlists may be conveniently stored as part of the data structure 134. The edit operation ends at 356.

Returning momentarily to the flow chart of Figure 9, if the user wishes to create a new playlist, the result of decision 316 is YES. In that event, the system executes processes illustrated in the flowchart of Figure 11 to create a new playlist. As previously discussed, the system 100 may simply display the titles of all musical tracks stored in the memory 104 and allow the user to manually select ones of the displayed musical tracks to add to the newly created playlist. Figure 11 illustrates

the operation of the system 100 to generate a playlist using metatags. In step 380, the user selects a desired metatag from the list shown, by way of example, in the screen display 250, illustrated in Figure 7. The user may select a metatag, such as genre, which causes the system 100 to display the display screen 252 listing the various genre metatags corresponding to the various musical tracks stored in the memory 104 (or in the storage device 126). In addition, as noted above, the user may select more than one metatag to further refine the selection of musical tracks. Thus, step 380 may represent multi-step processes in which one or more screen displays are provided to the user to guide the user through the metatag selection process.

After one or more metatags have been selected in step 380, the system 100 searches the data structure 134 (see Figure 1) in step 382. In one example implementation, the data structure 134 may be a conventional database in which search terms, such as the selected metatags, are provided as inputs to the database and results are produced by the database in the form of one or more musical tracks whose metatags correspond to the user-selected metatags.

In step 384, the system automatically adds to the playlist musical tracks whose metatags match the user-selected metatags. The automatically selected playlist is displayed for the user in step 386. The user may manually edit one or more of the musical tracks on the newly generated playlist in the manner described above with respect to the flowchart of Figure 10. Alternatively, the system 100 may simply display the resultant matches and permit the user to manually select which musical tracks will be added to the newly created playlist. In step 388, the completed playlist is stored in the memory 104 or, alternatively, in the data structure 134. The process ends at 390.

Thus, the system 100 provides a powerful but simple interface that allows the user to quickly generate playlists from stored musical tracks using one or more user-selected metatags. The system further provides simple editing processes that allow the user to readily alter existing playlists.

The system 100 has been described above primarily with respect to musical tracks since music generally imposes the greatest technical constraints on the portable digital audio device. However, the portable digital audio device is fully capable of playing other types of data, such as audio books or even video data. As noted above, different CODECs are designed specifically for data processing of particular data types. For example, MP3 is a widely used CODEC for music applications. However, other CODECs, such as adaptive delta modulation (ADM) were developed by NASA to produce intelligible speech at significantly reduced bandwidth and in the presence of high rates of bit rate errors. Other well known CODECs for speech include PCM, ADPCM and CELP. The CELP CODEC is one of the most commonly used CODECs for producing good quality speech that rates below 10,000 bits per second (Kbps).

As previously discussed, the display 108 may be a touch-sensitive display that allows control functionality to be programmed and operated by the user simply by touching the display 108 at a predetermined location. As one skilled in the art can appreciate, the control functions required for speech applications, such as an audio book, are significantly different from the control functions required for music data. For example, Figure 5 illustrates the use of the programmable control buttons 192, such as the fast forward button 206, rewind button 208, and the like, used to operate the portable digital device to play music files. In contrast, an audio book device may have programmable controls, such as "Next Page", "Previous Page", "Next Chapter", "Bookmark", and the like. It is highly desirable to alter the functionality of the touch-sensitive display 108 based on the type of data and the type of CODEC being used to process that data.

Figure 12 is a functional block diagram of a media interface system 400 to control the operation of the CODEC 114 (see Figure 2) and the touch-sensitive display 108. The media interface system 400 comprises an interface manager 402, a media interface manager (MIM) 404 and a CODEC manager 406. The interface manager 402 is responsible for loading "skins," which are the visible

portion of the system 100 that are viewed and operated by the user. The interface manager 402 is also responsible for displaying and updating controls on the touch-sensitive display 108 and sending messages about controls, such as button clicks, and the movement of the scroll bar 190 (see Figure 4) to the MIM 404.

5 The interface manager 402 controls the initiation of the execution for the portable audio device. Its functionality is similar to that of a windows scripting interpreter. That is, the interface manager 402 need only know how to operate the display 108 and need not know how to operate other portions of the system 100. While the interface manager 402 displays controls that are selected for proper operation of the selected CODEC 114, the interface manager has no inherent knowledge of their functionality with respect to a given CODEC. That is, the interface manager 402 simply reports status, such as user activation, of the defined controls on the touch-sensitive display 108 and displays interface updates, such as a selected track and a play list, or track time.

10 Just like a Window scripting interpreter, the interface manager 402 does not know what it means when it displays something or why it issues commands, it just recognizes when it is supposed to perform these behaviors. The interface manager 402 does not know what “pause” means when it tells a CODEC 114 to pause, it only knows that the user told it to carry out the “pause” command. It does not know how the CODEC 114 scans, it just relays this information on behalf of the user. It does not know how the track progress events will react to a pause command (they will stop updating), but rather it just communicates those events that it receives from the CODEC 114.

15 A list of controls that the interface manager 402 is to instantiate resides in a “skin” file stored in the memory 104. Each control in the skin file is described by the control type, its identification (ID), and a parameter list. For example, an entry in the skin file could describe a pushbutton control with an ID of “ID_PLAY_BUTTON” and a parameter list that details the location of the button on the touch-sensitive display 108 and a data file that contains a bitmap of the button’s

pressed image. If the user activates the play button, the interface manager 402 detects the activation.

A change in the state of a control causes the interface manager 402 to generate a message that is sent to the MIM 404. For example, if the user activates a play button by touching a stylus or finger over an area defined by a pushbutton on the touch-sensitive display 108, the interface manager 402 loads the bitmap file to illustrate the button in an activated or pressed position. The interface manager 402 also sends an interface command to the MIM 404 to inform the MIM of the button's ID and its pressed state.

In addition to user activation causing a state change that is detected by the interface manager 402, the MIM 404 can send data/commands to the interface manager 402 to thereby direct changes in the state of a control. For example, Figure 5 illustrates the elapsed time 230 of a musical track. The MIM 404 can respond to a frame progress update from the CODEC 114 (see Figure 1) and request that a text box labeled "ID_TRACKTIME" display the value "01:16." Thus, the touch-sensitive display 108 can be altered by the interface manager 402 in response to user activation or in response to commands from the MIM 404.

The task of the MIM 404 is to function as the "middleman" between the touch-sensitive display 108 (see Figure 1) and the CODEC 114. The touch-sensitive display 108, and the data contained therein, may be referred to generically as the user interface. The user interface is platform-dependent and must be designed with knowledge of the portable audio device's windowing style, menu system, file system, input/output (I/O), and the like. In contrast, the CODEC 114 is relatively platform-independent. That is, the CODEC 114 simply has a series of input controls, output status information, and accepts a stream of data for processing. The CODEC 114 need contain almost no information about the style of interaction for the portable audio device. Therefore, the task of the MIM 404 is to translate user interaction with the interface into commands transmitted to the CODEC 114 and to

translate status data and notifications sent from the CODEC into output to be displayed on the touch-sensitive display 108.

Although illustrated in Figure 12 as a single MIM 404, the system 100 typically instantiates a MIM 404 for each CODEC type. For example, the system 5 100 may contain a MIM 404 for audio CODECs such as MP3 and WMA, a MIM for video CODECs, such as MPEG-1 and AVI, a MIM for chaptered audio CODECs, such as CELP, and the like. For a specific CODEC to work in the system 100, it is only necessary for the CODEC to conform to the standards for that CODEC type. Thus, for each CODEC the corresponding MIM 404 relays the 10 minimum skin requirements for a given data file type to the interface manager 402, translates messages from the interface manager into messages for the CODEC 114, sends output data from the CODEC manager 406 to the interface manager 402 for display on the touch-sensitive display (*e.g.*, track time), provides specific menus that may be required on the touch-sensitive display 108 and provides a list of functions that can be mapped into buttons by the interface manager 402. The MIM 404 must 15 be capable of processing audio commands (*e.g.*, play/pause/stop, track time display, etc.).

A single MIM 404 can function satisfactorily with multiple CODECs if the MIM understands which CODEC functions to call and which CODEC output 20 data to process. In addition, a MIM 404 can work with any number of skins so long as the selected skin provides the controls needed by the MIM. Conversely, a skin can also be loaded by several different MIMs 404 as long as the skin provides the controls that are needed by the selected MIM. However, in a typical implementation, each skin is typically associated with a specific MIM 404 whose 25 functionality closely matches that of the skin. For example, a skin typically associated with a chapter book may be used to play a music file, but certain functionality, such as a “Next Chapter” button would not have any useful function. Accordingly, the skin is generally closely associated with a single MIM 404.

The skin may also contain a menu system for altering options in the CODEC 114. For example, the down-sampling rate may be user-controllable. In this event, the touch-sensitive display 108 may display the sample rate indicator 234 and allow the user to alter it by activating one or more of the buttons on the touch-sensitive display 108 in the manner previously described. In a typical embodiment, data concerning options that effect the operation of the CODEC 114 are contained within the MIM 404 since the interface manager 402 and the skin are ignorant of the CODEC and its functionality. Thus, these types of options are passed from the MIM 404 to the interface manager 402 to be added to the touch-sensitive display 108 (see Figure 1). Other functionality, such as loading files and play lists can be contained entirely within the interface manager 402 since these options are essentially independent of the CODEC 114.

The MIM 404 must also be aware of the capabilities of a particular CODEC 114 (see Figure 1) and pass any necessary information concerning the CODEC to the interface manager 402. For example, if a particular CODEC does not support down-sampling, the MIM 404 can disable that functionality in the menu. As previously discussed, the system 100 implements a number of different CODECs, one of which is selected as the CODEC 114 based on the type of data file. The CODEC manager 406 must determine the appropriate CODEC for a given file type and further, provide a set of command structures, sometimes referred to as "hooks," to the MIM 404 to allow access to the CODEC's functionality. The CODEC manager 406 can determine the appropriate CODEC for a given file type by taking a sample from the data file in attempting to decode it with one or more different CODECs. Depending on its ability to decode, the CODEC manager 406 determines that it either has a compatible rendering system (*i.e.* the CODEC 114 successfully translated the data) or that a particular data file cannot be rendered.

The MIM 404 also gains additional information about the processing capabilities of the CODEC 114 by asking or checking with the CODEC about its ability to process certain commands. For example, a particular CODEC 114 may be

used with multimedia data and requires processing video data and audio data in synchrony so that the video images and the sound are property synchronized. The MIM 404 receives information from the CODEC as to the properties that can be manipulated by the CODEC. Thus, the MIM 404 and the CODEC 114 function cooperatively such that the MIM 404 is aware of the CODEC capabilities. In turn, the MIM 404 cooperates with the interface manager 402 to provide the interface manager with the necessary information to implement the skin and required external functionality, such as control buttons on the touch-sensitive display 108.

The operation of the system 100 to match the CODEC 114 in the proper interface is illustrated in the flowcharts of Figures 13 and 14. At a start 410, illustrated in Figure 13, the system 100 is assumed to be under power. In step 412, the CODEC manager 406 (see Figure 12) reports the file type. That is, the CODEC manager 406 provides information to the MIM 404 indicating the type of data file requested by the user, such as a music data file or a speech data file. In decision 414, the system determines whether the current MIM 404 is satisfactory for the reported file type. As noted above, MIMs may be used with more than one CODEC. If the current MIM is satisfactory for the reported file type, the result of decision 414 is YES and, in step 416, the system 100 processes the data file in accordance with user-selected commands. This may include, by way of example, playing the data file.

If the currently selected MIM 404 (see Figure 12) is not satisfactory for the reported file type, the result of decision 414 is NO. In that event, in step 420, the system 100 checks the registry to identify the correct MIM 404 for the reported file type. As noted above, the CODEC 114 is matched with the MIM 404. When a new CODEC is installed in the system, the registry is checked to see if the matching MIM is already present on the portable audio device. If the correct MIM 404 is not installed, or if an older version of the MIM is present on the portable audio device, then the MIM (or updated version) is installed and the proper registry setting is

added. Whenever a MIM 404 is installed on a portable audio device, a compatible skin is typically installed as well.

In decision 422, the system 100 determines whether the registry contains the appropriate matched MIM for the reported file type and selected
5 CODEC 114 (see Figure 1). If the registry does not contain the appropriate match, the result of decision 422 is NO and, in step 424, the system displays an error message. As noted above, the corresponding MIM 404 is typically installed at the same time as a CODEC 114. Therefore, a match will generally be found in the registry. If the appropriate match is found within the registry, the result of decision
10 422 is YES and, in step 426, the system 100 unloads the old MIM and loads the new MIM.

After the new MIM is loaded, the system 100 determines whether the current skin is satisfactory in decision 430, illustrated in Figure 14. As previously noted, a single skin may be used with multiple MIMs. If the current skin is
15 satisfactory for the newly selected MIM, the result of decision 430 is YES and in step 432, the new MIM passes the appropriate menu items discussed above to the interface manager 402 (see Figure 12). If the currently selected skin is unsatisfactory for the new MIM, the result of decision 430 is NO. In that case, in step 434, the system 100 checks the registry for a compatible skin for the newly
20 installed MIM.

In decision 436, the system 100 determines whether an appropriate match has been found in the registry for a skin that is compatible with the newly installed MIM 404 (see Figure 12). If no match is found, the result of decision 436 is NO and, in step 438, the system displays an error message. As noted above, a
25 new skin is typically installed at the same time a new MIM is installed. Therefore, a match is generally found in the registry. If an appropriate match is found in the registry, the result of decision 436 is YES. In that event, in step 440, the system unloads the old skin and loads the new skin. Whenever a new skin is loaded, the skin is checked at a syntactic and symantec level to make sure that all of the controls

are properly specified. In step 442, the system 100 checks the controls for proper functionality.

In step 444, the interface manager 402 (see Figure 12) queries the MIM 404 for a list of required controls. As previously discussed, the list contains
5 each control's type and I.D. (e.g., type: PUSH_BUTTON, I.D.: ID_PLAY_BUTTON). If the skin cannot provide all of the required controls, then the user may be informed, via an error message, that the present skin is not valid for the reported file type. Alternatively, the MIM 404 can provide a list of controls that are required to support less than full functionality. If the skin has the required
10 minimum set of controls, the user can be asked if degraded functionality is acceptable. For example, a CELP file could be played on a skin designed for simple music files if the user is willing to forego options, such as chapter-hopping ability.

In step 448, the interface manager 402 loads the skin and creates the controls for display on the touch-sensitive display 108 (see Figure 1). Examples of
15 the skin and controls are illustrated in the screen displays of Figures 3-8. In step 450, the interface manager 402 passes "handles" to the MIM 404. As those skilled in the art can appreciate, the handles are generally in the form of a data structure containing parameter lists that are used by the MIM 404 to determine which buttons are present on the touch-sensitive display 108 and when a state changes, such as
20 when a button is activated by the user. The process of initialization ends at 452.

At this stage, the appropriate skin has been loaded such that the touch-sensitive display 108 (see Figure 1) contains all of the appropriate controls and display format that corresponds to the particular file type selected by the user. In addition, the appropriate CODEC 114 has been selected for the reported file type.

25 The MIM 404 detects state changes in the touch-sensitive display 108, as reported by the interface manager 402, and converts state changes into commands for the CODEC 114. The CODEC manager 406 receives the CODEC commands and passes those commands along to the CODEC 114. The CODEC manager 406 also provides data feedback from the CODEC 114, such as track time for eventual

display on the touch-sensitive display 108. The MIM 404 translates such data into the appropriate structure and format required by the interface manager 402 and passes the information to the interface manager. Thus, the system allows for the automatic selection of CODEC and appropriate matching interface that is mapped to that CODEC. Thus, the user is automatically provided with the most appropriate interface and control set for each file type available on the portable digital audio device.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. For example, the operation of the system 100 has been described using the example of musical tracks as the audio data files that are selected by a user and placed in playlists. However, the system 100 is applicable to any type of audio data file, such as audio books, as well as musical data files. Accordingly, the invention is not limited except as by the appended claims.